FAST AND ROBUST CLOTH SIMULATION

A Thesis
Presented to
The Academic Faculty

by

Michael Y. Shin

In Partial Fulfillment
of the Requirements for the Degree
BS in Computer Science in the
School of Computing

Georgia Institute of Technology
December 2012

FAST AND ROBUST CLOTH SIMULATION

Approved by:

Dr. Liu, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. Turk, Second Reader
School of Interactive Computing
Georgia Institute of Technology

Dr. Bruckman, Undergraduate Research Coordinator
School of Interactive Computing
Georgia Institute of Technology

Date Approved: [Date Approved by Committee]

TABLE OF CONTENTS

		Page
LIST OF FIGURES		iii
SUMMARY		iv
<u>CHAPTER</u>		
1	Introduction	1
2	Methods	3
3	Limitations	5
4	Results	6
APPENDIX:	Statistics	8
REFERENCES		9

LIST OF FIGURES

Page

Figure 1: Baraff and Witkin's Implicit Integrator 2

Figure 2: Simulator and coefficient adjuster 6

Figure 3: Simulated cloth at start, after 20 iterations, and after 40 iterations in rest state 6

SUMMARY

The goal of this project is to develop a realistic, robust, and fast cloth-simulator - realistic and robust, so that the clothes behave as expected; and fast, so that results can be seen instantly and the clothes can be interacted with in real-time. A variety of papers on previous works are referenced, so that the specific advancements of each research may be combined to produce the desired simulator. The long-term intention of this project is to serve as the requisite physical engine in which a protocol for the animation of digital characters putting on clothes may be developed. This has applications in robotics, wherein the protocol used by the digital character can be taught to robots, which may then help people who have difficulty dressing themselves.

CHAPTER 1

INTRODUCTION

The bases for current physics-based cloth simulation were established more than two decades ago, when Terzopoulos et al. modelled cloth simulation in terms of mechanical engineering and the finite element method [Terzopoulos et al. '87]. Ever since, other researchers built upon this foundation by experimenting with many different methods to maximize performance (speed of simulation) and realism in cloth simulation. Although newer methods and better hardware generally bring improvements, each different method has its own distinct balance between realism and performance, as one of the two is generally traded off for another. For example, Breen and others developed a way to produce very realistic images of cloth draped on objects [Breen et al. '94]. However, the process is very slow, because the focus of this specific method is primarily on realism over performance. Each newer method also attempts to address a very specific shortcoming in the previous methods. For example, Choi and Ko's method of simulation attempts to make the buckling and the bending of simulated cloth appear more realistic [Choi and Ko '02].

There are general, universal components in cloth simulation: integration, cloth structure, forces-modelling, collision detection and handling, etc. Each new simulator, even while focusing on a specific improvement for a component, also tries to make improvements or variations in the other components, usually as a side-improvement or to adjust the balance between performance and realism to better showcase the primary novel improvement.

The proliferation and the variety of the methods for simulating the different components of cloth simulation offer many choices of methods for a new cloth simulator.

I can thus pick and choose the best method for each component of cloth simulation, in order to compose a robust (non-glitchy and realistic) simulator that runs in real-time.

CHAPTER 2

METHODS

Baraff and Witkin's simulation uses an implicit integrator (shown below) to ensure robustness in a fast simulation with large time steps, and their constraint-energy based model of deriving the different internal forces associated with cloth (stretch, shear, and bend) works well with their integration method [Baraff and Witkin '98]. The following equation is derived from the formula relating force to mass and acceleration. Unlike forwards explicit Euler method, the backwards implicit Euler method ensures that even with a large time step, the displacements correspond with the equation for the state at the end of the time step.

$$\left(\mathbf{I} - h\mathbf{M}^{-1}\frac{\partial \mathbf{f}}{\partial \mathbf{v}} - h^2\mathbf{M}^{-1}\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)\Delta\mathbf{v} = h\mathbf{M}^{-1}\left(\mathbf{f}_0 + h\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{v}_0\right)$$

Figure 1: Baraff and Witkin's Implicit Integrator

In a system with *n* vertices,

I is a 3*n*-by-3*n* identity matrix.

h is the scalar time step.

M is a diagonal 3*n*-by-3*n* matrix. Each group of 3 consecutive values along the diagonal is the mass of each particle.

δf/δx is a 3*n*-by-3*n* matrix derivative of the force on *n* vertices with respect to their positions.

δf/δv is a 3*n*-by-3*n* matrix derivative of the force on *n* vertices with respect to their velocities.

Δv, the unknown to be solved for, is the 3*n* vector of the change in the vertices' velocities for the current iteration.

f_0 is the 3*n* vector of the net force acting on the *n* vertices.

v_0 is the 3*n* vector of the velocities of the vertices in the current iteration.

Whereas Baraff and Witkin's research deals primarily with the implicit integrator, Bridson et al.'s research details a robust method for the treatment of collisions [Bridson et al. '02]. As per its recommendation, an axis-aligned bounding box hierarchy is implemented to eliminate unnecessary checks for collisions between triangles. Provot describes a way to take advantage of the hierarchical structure to efficiently detect patches of cloth that need not to be checked for self-collision [Provot '97]. By extracting curvature information from the surface normals of the mesh, one can assume that cloth is not distorted enough to touch itself if there is very low curvature in the area. Although this is not robust to all conditions, this simplification did not cause problems for Provot's clothing simulation.

CHAPTER 3

LIMITATIONS

Bending force is not implemented in the simulator.

Without reference, it is difficult to figure out what force coefficients ensure the greatest stability and realism. Default values are set according to results from trial-and-error experience.

The simulation is not robust to extreme conditions. When coefficient values are made extreme, or when the mesh is stretched or moved about, the system blows up. It remains to be seen whether the system will remain stable with the correct coefficients, and without extreme outside forces. Collision handling algorithm takes shortcuts on the assumption that clothing models are convex, and similarly are not extraordinary in their formation.

CHAPTER 4

RESULTS

So far, the simulator simulates a 100-particle system around real-time. Although this is without the implementation of the bending force, the current speed means that the process of solving a large system of equations expressed as sparse matrices is quite fast. The effect of the implementation of one more force should be trivial. The chart in page 8 shows that performance does not have a linear relation with the number of particles in the system.

Coefficients of 50000 and 5 for the stretch force and its damping force, and 100 for shear force were used to produce a nice simulation for this particular system. After a certain threshold around 10000, the increase of the stretch force coefficient seemed to have very little effect. It seems that this value is ideal for keeping the cloth resistant to stretch, without risking instability from extreme coefficients.

The implementation of adjustable coefficients within the simulation's user interface facilitated the testing of different coefficients on stock systems. The goal is to find the ideal coefficients and the ideal system sizes that will produce the most realistic, yet efficient, simulation.

Although Baraff and Witkin's simulations were not real-time ("Representative running times include a long skirt with 4,530 nodes (8,844 triangles) on a dancing character at a cost of 10 seconds per frame, and a shirt with 6,450 nodes (12,654 triangles) with a cost varying between 8 to 14 seconds per frame ") [Baraff and Witkin

'98], processors have since become faster, and the simulator is expected to be used with much smaller systems, as its focus in on realistic behaviour, not appearance.

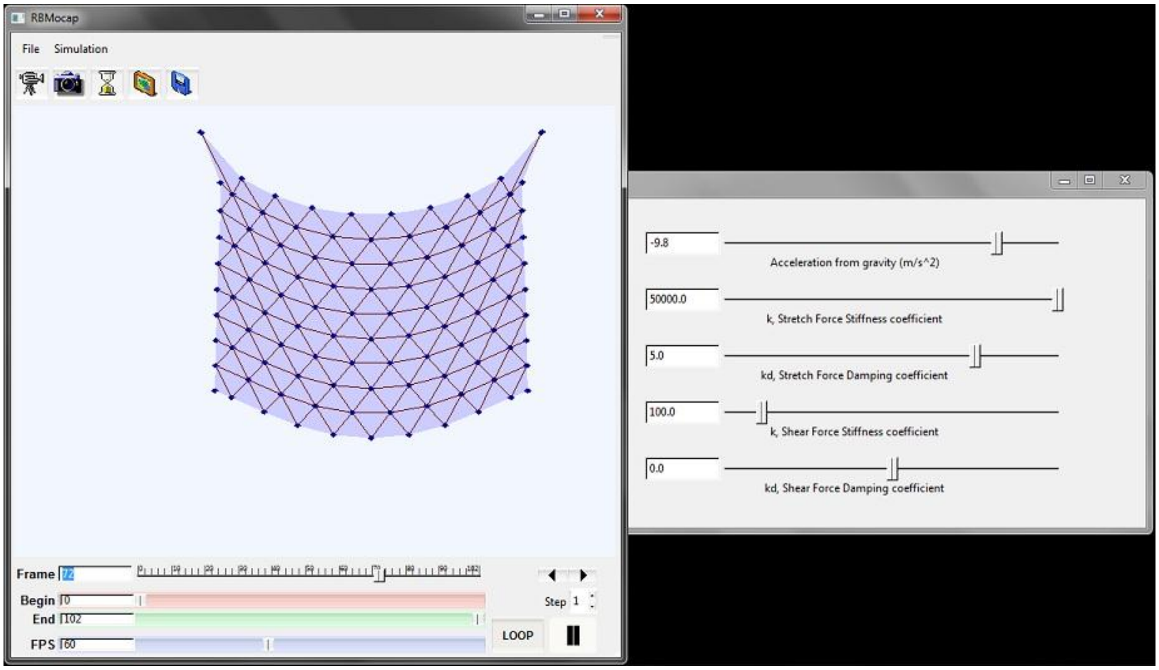


Figure 2: Simulator and coefficient adjuster

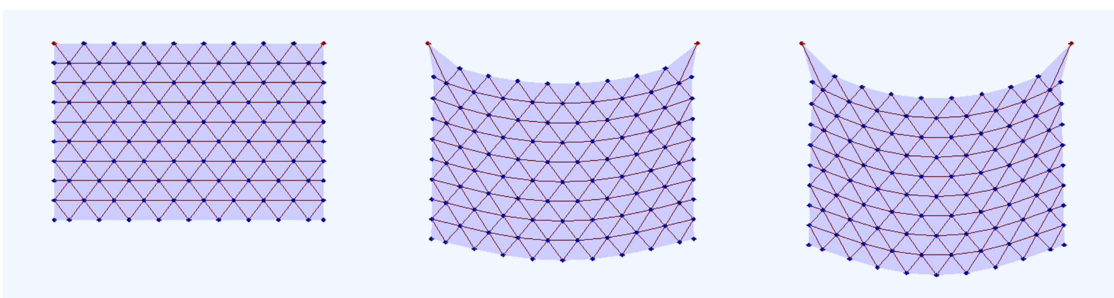
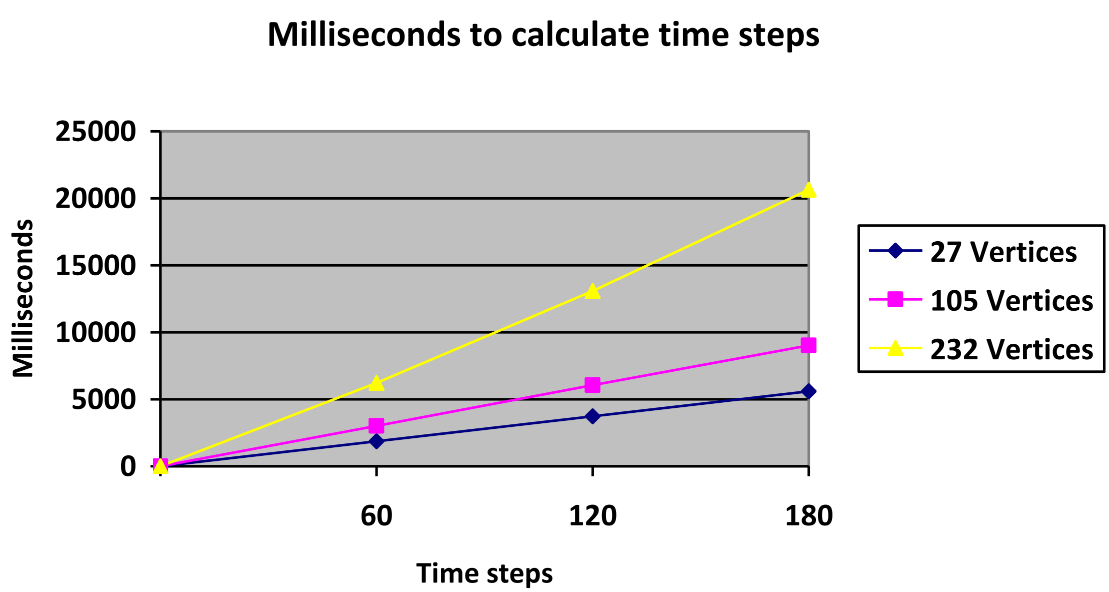


Figure 3: Simulated cloth at start, after 20 iterations, and after 40 iterations in rest state (approximately 1 second and 2 seconds after start, respectively)

APPENDIX

CHARTS

Vertices in system	Milliseconds to calculate 60 time steps	Milliseconds to calculate 120 time steps	Milliseconds to calculate 180 time steps
27	1862	3734	5606
105	3018	6051	9010
232	6219	13097	20642



REFERENCES

D. Baraff and A.P. Witkin. Large Steps in Cloth Simulation. *Computer Graphics (Proc. SIGGRAPH)*, pages 43-54, 1998.

D.E Breen, D.H. House, and M.J. Wozny. Predicting the drape of woven cloth using interacting particles. *Computer Graphics (Proc. SIGGRAPH)*, pages 365-372, 1994.

R. Bridson, R. Fedkiw, and J. Anderson. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. *ACM Transactions on Graphics*, 21(3), pages 28-36, 2002.

K. Choi and H. Ko. Stable but Responsive Cloth. *ACM Transactions on Graphics*, 21(3), pages 604-611, 2002.

X. Provot. "Collision and Self-Collision Handling in Cloth Model Dedicated to Design Garments." Proceedings of Graphics Interface, pp. 177-189, 1997.

D. Terzopoulos and K. Fleischer. Deformable models. *Visual Computer*, 4:306-331, 1988.

D. Terzopoulos and K. Fleischer. Modeling inelastic deformation: Viscoelasticity, plasticity, fracture. In *Computer Graphics* (*Proc. SIGGRAPH*), volume 22, pages 269–278. ACM, August 1988.

D. Terzopoulos, J.C. Platt, and A.H. Barr. Elastically deformable models. *Computer Graphics* (*Proc. SIGGRAPH*), 21:205–214, 1987.